REMARKS

Claims 1-9 and 14-20 remain pending for the present application. Applicants respectively request reconsideration of the above referenced application in view of the above claim amendments and the remarks presented below.

35 U.S.C. Section 112 rejections:

Paragraphs 2-6 of the above referenced Office Action reject Claims 14 and 15 as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possess on of the claimed invention. Applicants respectfully traverse.

Applicants point out that the present invention as recited in independent Claims 1,3, and 8 comprises a translation process that responds to an exception during execution of a stored translated instruction by rolling back to a point in execution at which correct state of a target processor is known. Once the rollback has been implemented, target instruction are interpreted in order from the point in execution at which correct state of a target processor is known.

One especially useful embodiment records data relating to the number of times a target instruction is executed at points which branches occur. Based on statistics regarding the number of times a particular branch has been taken, that particular sequence of instructions may be speculatively considered to be a super-block. The sequence may be translated, optimized, and linked. The translated optimized, linked result is stored as such in the translation cache. If the speculation is true, then significant execution time is saved. If not, an exception is taken which returns the code morphing software to the interpreter.

Claim 14 adds quotations reciting the exception results from speculative execution of a branch instruction of a target application. Applicants point out that speculative execution is one core feature of the present invention and is explicitly described in the specification. For example, in the specification, the paragraph beginning on from page 13, line 19, and ending on, page 14, line 10, explicitly recites:

* * *

Moreover, if the interpreter is utilized to collect statistics in addition to the number of times a particular target instruction has been executed, additional significant advantages may be obtained. For example, if a target instruction includes a branch, the address of the instruction to which it branches may be recorded along with the number of times the branch has been executed. Then, when a number of sequential target instructions are executed by the interpreter, a history of branching and branch addresses will have been established. From this, the likelihood of a particular branch operation taking place may be determined. These statistics may be utilized to guide super-block formation. By utilizing these statistics, a particular sequence of instructions may be speculatively considered to be a super-block after being executed a significant number of times. After being interpreted for the selected number of times, the sequence may be translated, optimized, linked through the various branches without the necessity to go through a separate linking operation, and stored as such in the translation cache. If the speculation turns out to be true, then significant time is saved in processing the instructions. If not, the operation causes an exception which returns the code morphing software to the interpreter.

* * *

Applicants point out that at least one widely used definition of "speculative execution" in the computer science field refers to speculative execution as being the execution of code whose result may not actually be needed.

Accordingly, applicants respectfully point out that "the exception results from speculative execution of a branch instruction of the target application" as recited in Claim 14, is explicitly supported by the above cited paragraph (at least).

Additionally, applicants respectfully point out that "speculatively translating target instructions into host instructions based on a likelihood of a branch being taken" as recited in claim 15, is explicitly supported by the above cited paragraph (at least). Based on

determined likelihood (e.g., statistics), a particular sequence of instructions may be speculatively translated, optimized, and linked, and the result stored as such in the translation cache.

Accordingly, Applicants point out the present invention as explicitly recited in Claims 14-15 is definite within the meaning of 35 U.S.C. Section 112.

35 U.S.C. Section 132 rejection:

Accordingly, no new matter is entered by the previously added Claims 14 and 15 and there is no valid 35 U.S.C. Section 132 rejection.

35 U.S.C Section 102 Rejections:

Claims 1-9 are rejected under 35 U.S.C. Section 102(e) as being anticipated by Krishnaswamy (U.S. Patent No. 6,308,318). Applicants respectfully traverses.

Applicants point out that the present invention as recited in independent Claims 1,3, and 8 comprises a translation process that responds to an exception during execution of a stored translated instruction by rolling back to a previous point in execution at which correct state of a target processor is known (emphasis added). Once the rollback has been implemented, target instruction are interpreted in order from the point in execution at which correct state of a target processor is known.

Applicants point out that this is completely different from the Krishnaswamy reference, which discloses a mechanism whereby code is executed from an exception point up to a point where a guaranteed application state recoverable point occurs. In other words, code continues to execute after an exception is received. Accordingly, Applicants

respectively assert that the claimed invention is not anticipated by Krishnaswamy within the meaning of 35 U.S.C. Section 102(e).

Additionally, Applicants point out that the exceptions disclosed in Krishnaswamy are different than the exceptions of the claimed invention. The Krishnaswamy reference describes the exceptions as being asynchronous exceptions, for example, from an external event. The exceptions of the claimed invention do not have to be asynchronous exceptions. The exceptions of the claimed invention do not have to be from external events. Applicants respectively submit that these limitations are not shown or suggested by Krishnaswamy.

35 U.S.C Section 103 Rejections

Claims 6-7 are rejected under 35 U.S.C. Section 103 as being rendered obvious by Krishnaswamy (U.S. Patent No. 6,308,318) in view of Lethin. Applicants respectfully traverse. Applicants point out that the present invention as recited in independent Claims 1,3, and 8 comprises a translation process that responds to an exception during execution of a stored translated instruction by rolling back to a previous point in execution at which correct state of a target processor is known. For the same rationale as described above, Applicants submit that these limitations are not shown or suggested by Krishnaswamy. Accordingly, the claimed invention is not rendered obvious by Krishnaswamy within the meaning of 35 U.S.C Section 103.

CONCLUSION

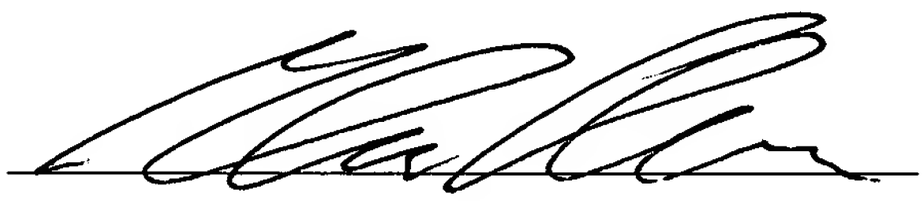
The Examiner is urged to contact Applicants' undersigned representative if the Examiner believes such action would expedite resolution of the present Application.

Please charge any additional fees or apply any credits to our PTO deposit account number: 23-0085.

Respectfully submitted,

WAGNER, MURABITO & HAO, LLP

Dated: 4/16/, 2004



Glenn Barnes
Registration No. 42,293

Two North Market Street
Third Floor
San Jose, CA 95113
(408) 938-9060